



项目3

创建第一个App

3.1 项目目标：用 Android Studio 创建 App

使用 Android Studio 可以十分便捷且以全程可视化的方式完成 App 的创建、运行与调试。相较于其他 Android 开发工具,响应速度更快、UI 主题更具设计性、调试程序更加智能等优势让 Android Studio 表现出更加优秀的属性。因此,使用 Android Studio 进行 App 的创建与设计将是一件相当快乐的事情,请大家保持愉快的心情开始 App 之旅。

3.2 项目准备

3.2.1 Android 的内部结构

为了保证 Android 程序结构的一致性,Android Studio 为每一个程序设置了相同的内部结构,该结构在 Android 项目建立之初就已经存在了。程序的内部结构是引导程序运行

及应用的向导,也是程序员在进行程序编写与设计时需要掌握与熟悉的内容。因此,对程序的内部结构的介绍是必不可少的。这里以创建名为 AndroidHello 的项目为例进行介绍。大家可以发现一个 App 程序是由多个文件及文件夹共同组成的,每个文件或文件夹都有不同的意义和功能。

在新建的 App 工程文件中,Android Studio 会自动生成许多文件,如图 3-1 所示。其中,重要的文件如下。



图 3-1 AndroidHello 项目的程序内部结构

(1) app: 在 Android Studio 中进行编程时,一般分为 Project(工作空间)、Module(模块)两种概念。app 为创建项目时默认模块,即一个 Module、一个 Android 应用程序的文档结构。

(2) libs: 用于存放项目的类库,例如项目中会用到的.jar 文件等。

(3) src: 用于存放该 Android 项目中用到的所有资源文件,例如图片等。

(4) androidTest: 用于存放应用程序单元的测试代码。

(5) main: Android 项目的主目录,其中 java 目录存放.java 源代码文件,res 存放资源文件,包含图像、字符串资源等,AndroidManifest.xml 是项目的配置文件。

- (6) build.gradle: Android 项目的 Gradle 构建脚本。
- (7) build: Android studio 项目的编译目录。
- (8) gradle: 用于存放该项目的构建工具。
- (9) External Libraries: 用于显示该项目所依赖的所有类库。

3.2.2 Android 的开发流程

对 Android 平台上的应用进行开发,可以按照如下流程来进行。

- (1) 安装 Android 调试软件,配置开发环境。
- (2) 创建 Android 虚拟机或硬件设备。
- (3) 创建 Android 项目,编写代码,提供资源文件。
- (4) 运行 Android 应用程序,用 Android Studio 运行程序并呈现效果。
- (5) 调试 Android 应用程序,测试并发布。

3.3 项目运行

3.3.1 创建一个 App

步骤 1: 双击 Android Studio 的图标打开工具,如图 3-2 所示。



图 3-2 打开 Android Studio 时的加载画面

步骤 2: 单击 Start a new Android Studio project 按钮, 创建第一个 Android 项目, 如图 3-3 所示。



图 3-3 创建 Android 项目

步骤 3: 为新建项目设置属性。其中, 项目的名称为 HelloWorld, 设定项目位置, 然后单击 Next 按钮, 如图 3-4 所示。

步骤 4: 选择适当的 SDK 版本, 然后单击 Next 按钮, 如图 3-5 所示。

步骤 5: 为 App 的运行选择一个 Activity 样式, 例如 Empty Activity, 然后单击 Next 按钮, 如图 3-6 所示。

步骤 6: 设置 Activity 的名称和 Layout 的名称, 然后单击 Finish 按钮, 完成新建程序的前期设定, 如图 3-7 所示, 继而进入 Android Studio 的开发主界面, 如图 3-8 所示。

3.3.2 运行 App

1. 使用模拟器运行 App

为了避免系统报错, 建议大家在创建模拟器之前单击 SDK Manager 按钮, 更新 Android SDK 的相关配置, 如图 3-9 所示。

步骤 1: 单击 AVD Manager 按钮, 弹出 Android Virtual Device Manager 窗口, 如图 3-10 所示, 然后单击 **+ Create Virtual Device...** 按钮, 创建虚拟设备模拟器。

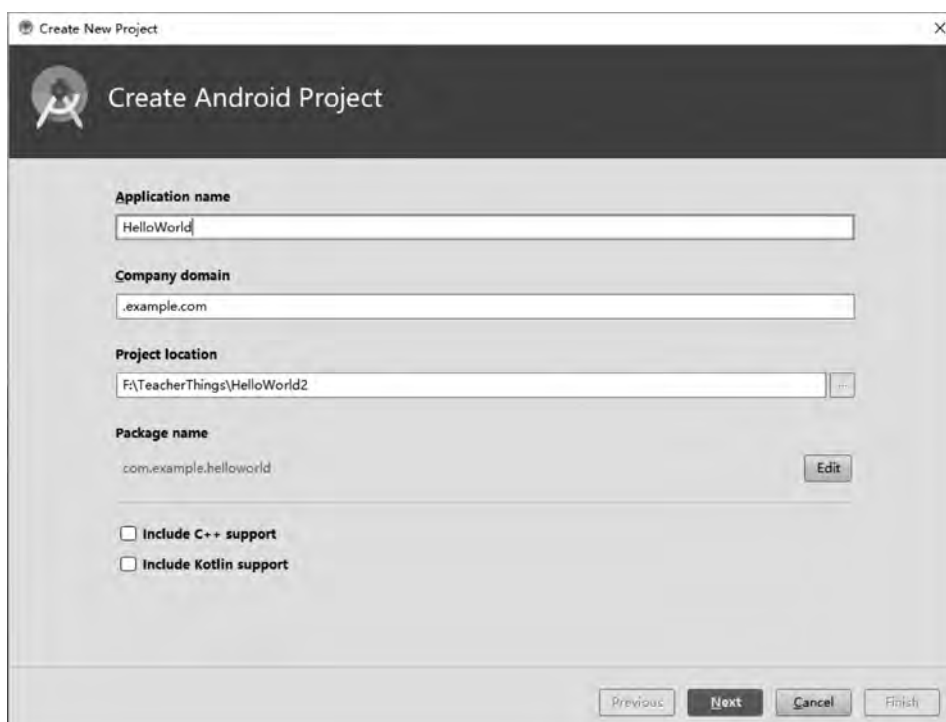


图 3-4 为新建项目设置属性

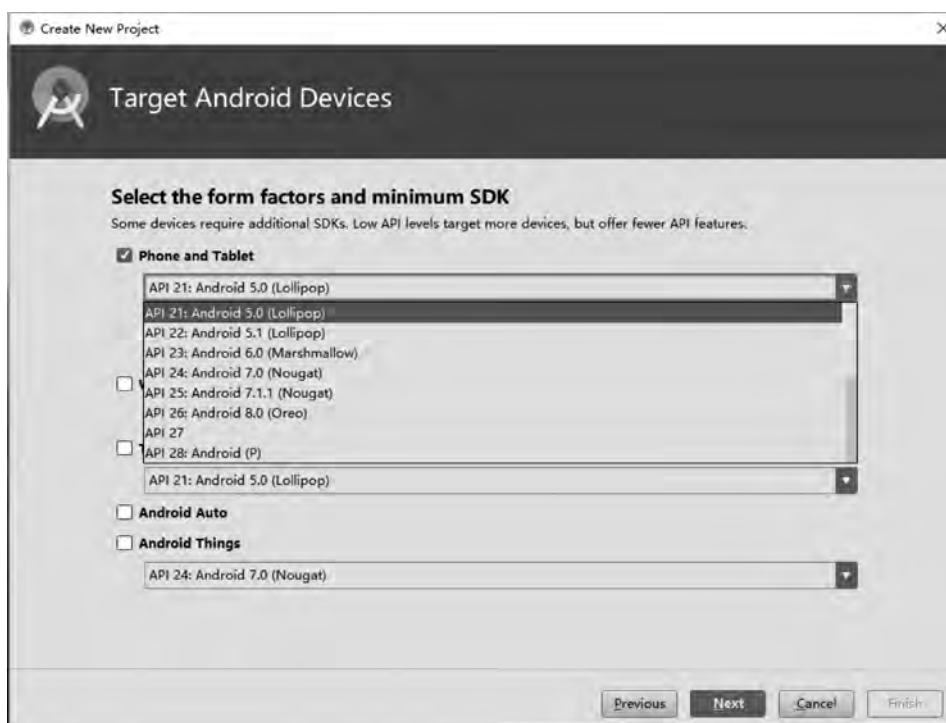


图 3-5 选择 SDK 版本



图 3-6 设置 App 的 Activity 样式

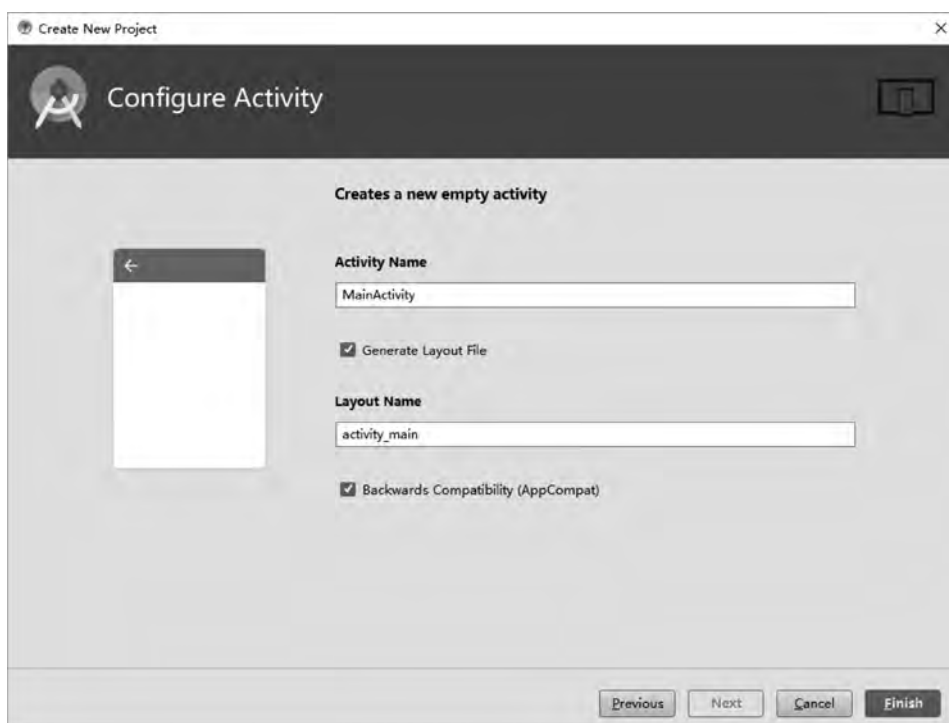


图 3-7 为新建 Activity 设置文件名称

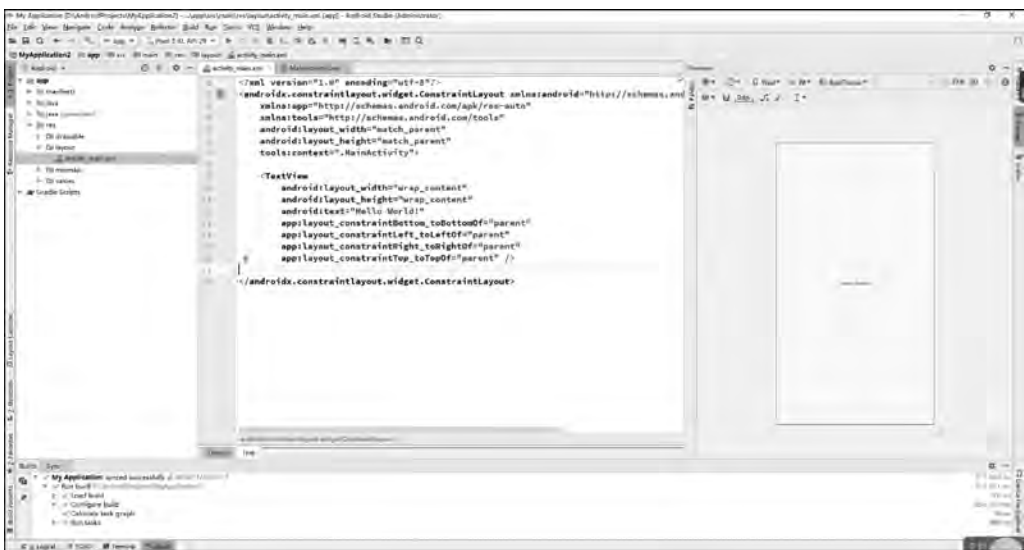


图 3-8 Android Studio 的开发主界面

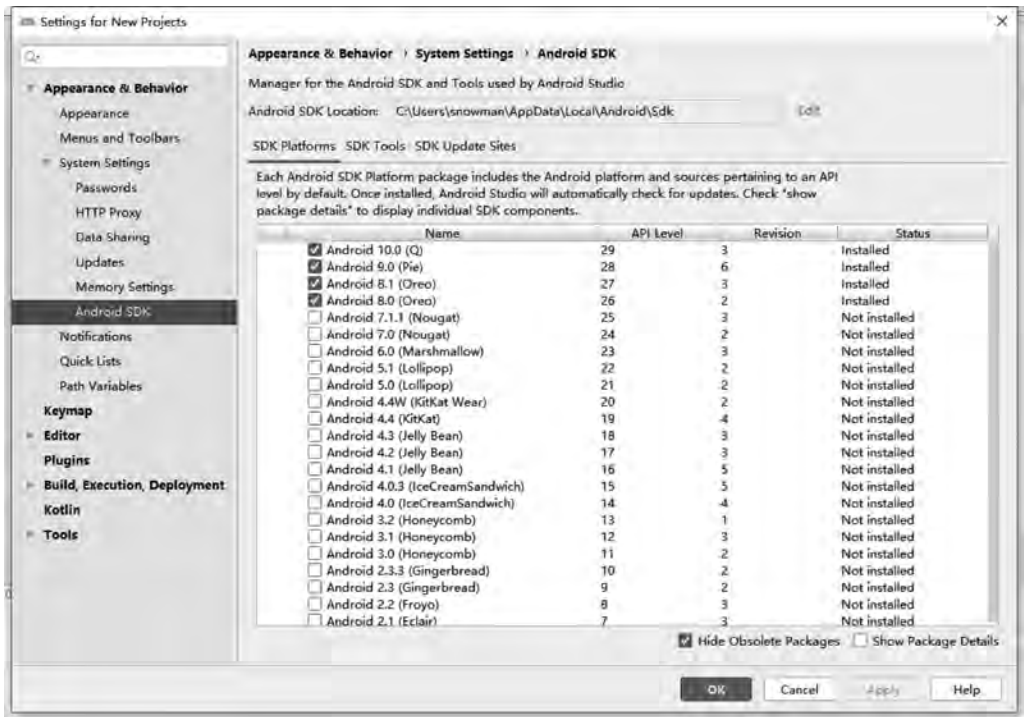


图 3-9 更新 Android Studio 的相关配置

步骤 2: 选择一种模拟器设备,如图 3-11 所示。

步骤 3: 确定设备后单击 Next 按钮,弹出 System Image 界面,选择系统版本,一般默



图 3-10 Android Virtual Device Manager 窗口

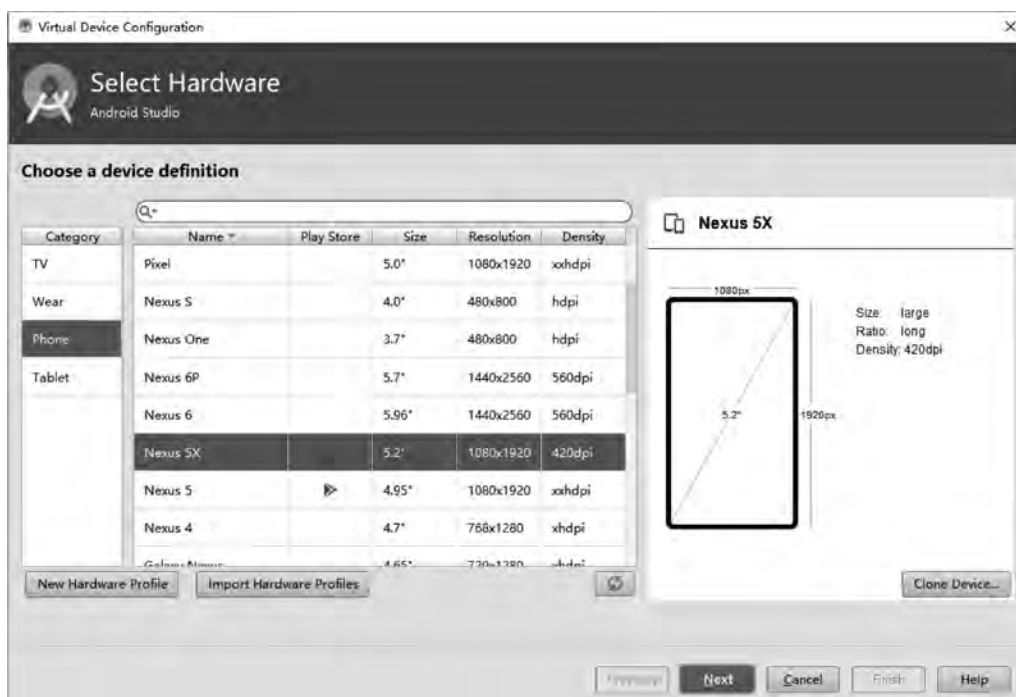


图 3-11 选择模拟器设备

认选择最高版本,如图 3-12 所示。

步骤 4: 单击 Next 按钮,弹出 Verify Configuration 界面,一般应用默认设置。需要注

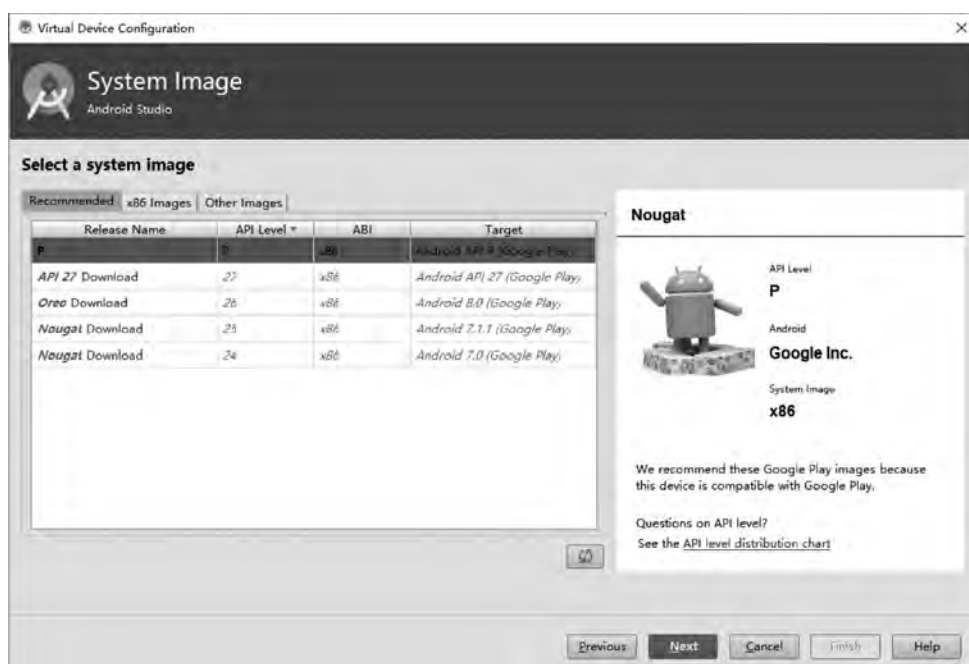


图 3-12 选择系统版本

意手机模拟器外观，一般将其设置为 Automatic Emulated Performance，即自动配置大小，如图 3-13 所示。然后单击 Finish 按钮。



图 3-13 AVD Verify Configuration 界面

步骤 5：加载一段时间后,模拟器即被创建出来,如图 3-14 所示。



图 3-14 新创建的模拟器

步骤 6：单击 Run App 按钮,即可弹出选择部署目标 Select Deployment Target 窗口,再选择可用虚拟设备 Available Virtual Device,如图 3-15 所示。

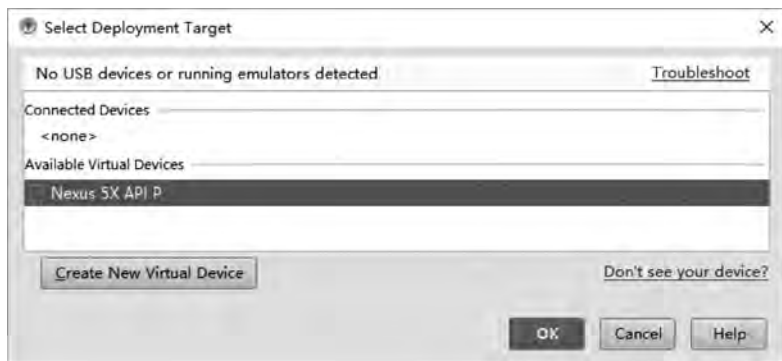


图 3-15 可用模拟器设备界面

步骤 7：单击 OK 按钮运行 App,可以通过 Android Studio 底部显示的情况来监视运行状态,加载一段时间后,即可出现使用模拟器运行的效果,如图 3-16 所示。

2. 使用手机运行 App

步骤 1：使用 USB 的方式将手机与计算机连接。

步骤 2：在手机上选中“连接后启动调试模式”,如果是首次连接,计算机会弹出安装对话框进行匹配安装。

步骤 3：在 Android Studio 的工具栏上单击“运行”按钮 ,选择部署目标 Select Deployment Target 窗口,选择要连接的设备,然后单击 OK 按钮,如图 3-17 所示。

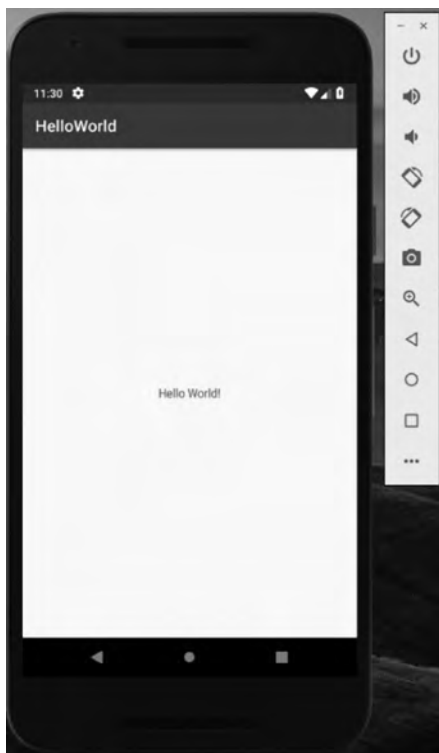


图 3-16 使用模拟器运行的效果



图 3-17 连接设备

步骤 4: 等待运行,即可在手机界面上看到 App 的演示效果,如图 3-18 所示。

3.3.3 调试 App

在使用 Android Studio 开发 App 的过程中,程序人员可能会由于各种原因在编程时产生错误或疏忽,因此通过调试的方法找到错误所在并进行修改是一项十分重要的工作。在使用 Android Studio 进行 App 开发时,可以采用断点调试的方法,具体操作过程如下。

步骤 1: 在 Android Studio 中找到需要调试的程序,如图 3-19 所示。

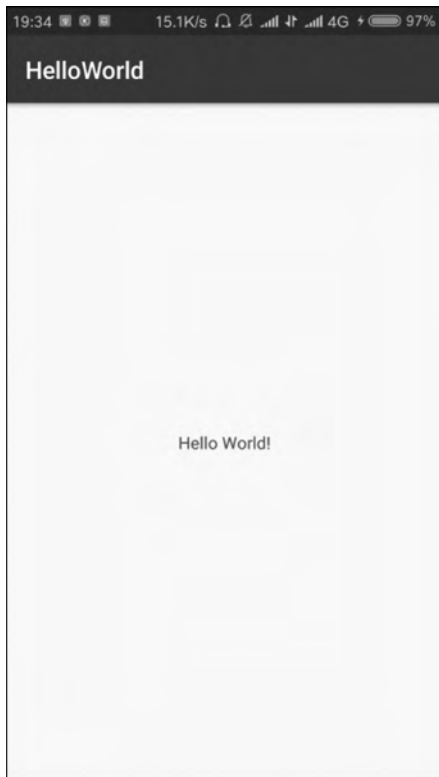


图 3-18 使用手机运行 App 的效果

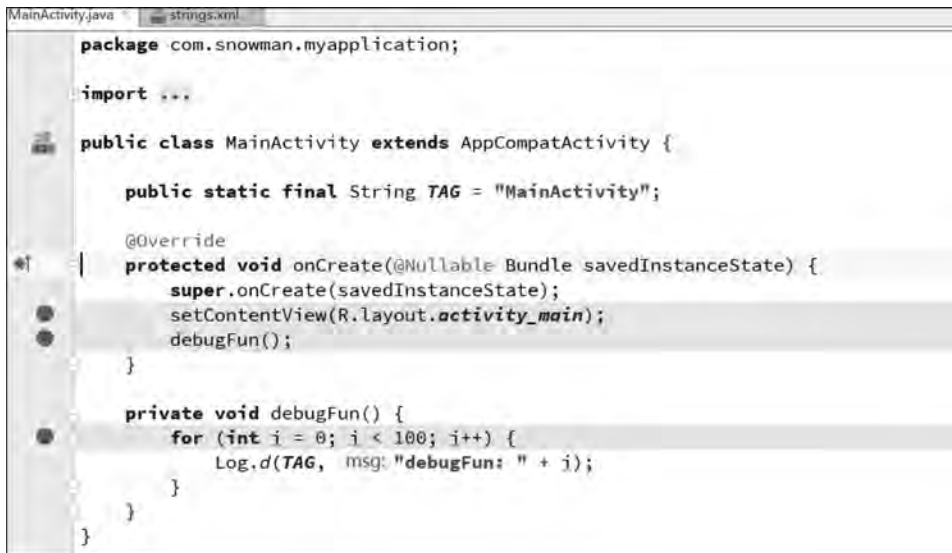


图 3-19 选择需要调试的程序

步骤 2：设置断点，即通过在行号处单击的方式进行设定。再单击  按钮，开启调试会话，在 Debug 视图中即可得到调试的情况，如图 3-20 所示。

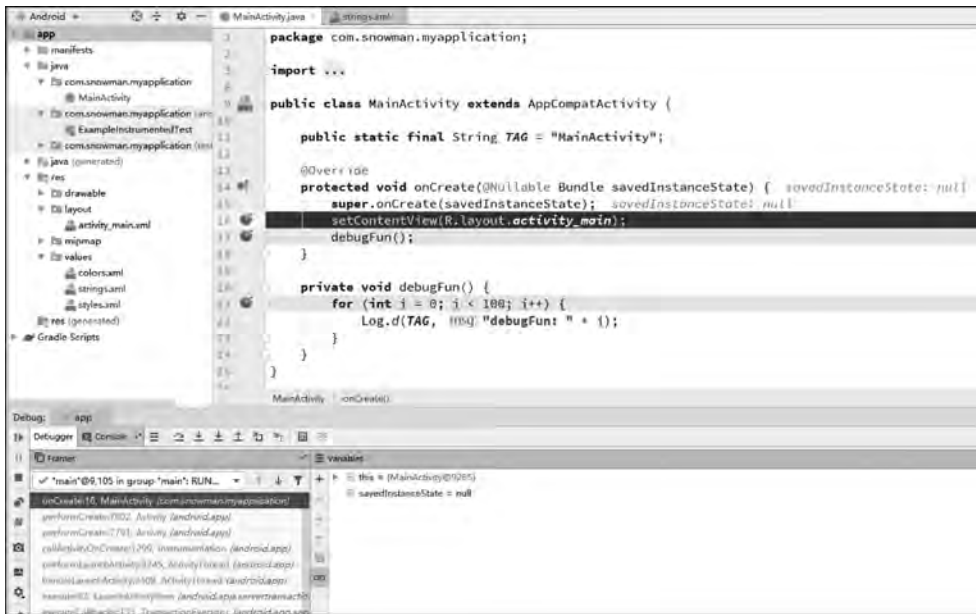


图 3-20 设置断点并开启调试会话

3.4 项目结案

本项目通过对 Android 内部结构的解析,让大家能够较为深入地体会到 App 的内部组成及彼此之间的关系;然后通过对 App 开发流程的梳理,明确开发 App 的全部步骤。从创建一个项目到运行项目,再到调试项目,整个流程紧密相关,缺一不可,为 App 的顺利编写提供了全面的技术保障。虽然在本项目中还没有接触到具体的编写方法,但是整体的流程与思路是大家需要掌握与熟练应用的。

3.5 项目练习

1. 创建一个名为“我的第一个 App”的小程序,体验完整的开发流程。
2. 在上一个小程序的基础上修改显示字符串为“Hello App”。
3. 在上一个小程序的基础上修改字符串的布局位置至屏幕左侧。